

Patent Claims

1. A method for visualizing and processing a value assembly process, the value assembly process being visualized as a set of value assembly lines (VAL), which value assembly lines are arranged on a number of different hierarchical levels, the value assembly lines having the following essential properties:
- each value assembly line has precisely one output interface and at least one input interface;
 - the value assembly line receives input value packages (IVP) via the input interfaces;
 - the input value packages are combined in the value assembly line in accordance with rules defined in a specific main line function (MLF), a value contribution being made, and the value added package (VAP) being generated;
 - the value added package is made available via the output interface; and
 - similar types of information are contained in the input value packages and in the value added package;
- the process visualization having the following basic properties:
- an uppermost hierarchy level (N) has precisely one value assembly line (VAL.N) of the highest hierarchy level, which generates a value added package (VAP.N);
 - the value assembly line of the highest hierarchy level receives input value packages (IVP) via its input interfaces from lower-order value assembly lines (SAL.N);
 - when a lower-order value assembly line (SAL.N) is focused on, it is visualized in an entirely similar way as a value assembly line (VAL.N-1) which receives its input value packages from lower-order value assembly lines (SAL.N-1) and

0483064-06101

- 5 - on each hierarchy level (M) down to a lowermost hierarchy level it is possible in each case to focus on a lower-order value assembly line (SAL.M) of this hierarchy level, which is visualized in a similar way as a value assembly line (VAL.M-1) of the next lower hierarchy level (M-1), which likewise receives input value packages via input interfaces, combines these, makes a value contribution, and makes a value added package (VAP.M-1) available at the output interface;
- 10 in such a way that the process is visualized as a fractal process in the case of which the structure of all value assembly lines is similar on all hierarchy levels,
- 15 value packages being processed in accordance with the following steps:
- on a lowermost hierarchy level, value packages (IVP) are supplied to the value assembly lines (VAL) of the lowermost hierarchy level across the system boundaries of the value assembly process under consideration;
 - the input value packages (IVP) of a lowermost hierarchy level are combined in value assembly lines (VAL) of this lowermost hierarchy level in accordance with its main line function (MLF), the value of the value packages is increased by a value contribution of the value assembly line and/or of the main line function, and a value added package (VAP) is made available at the output interface of the value assembly line;
 - on all hierarchy levels up to a highest hierarchy level, the value added package (VAP) is passed on to precisely one value assembly
- 20
- 25
- 30
- 35

line (VAL) of the next higher hierarchy level, and serves this value assembly line as input value package;

- 5 - the value flows take place strictly in one direction, in each case from a lower hierarchy level into a higher hierarchy level, and the value assembly lines of a hierarchy level are not interconnected.

10 2. The method as claimed in claim 1, in which each value added package (VAP) is compared with a reference value added package (RVAP), and in which impermissible deviations of the value added package (VAP) and of the reference value added
15 package (RVAP) are detected and reported via a warning function.

20 3. A Computer program for simulating and illustrating a value assembly process, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which value assembly process value packages are transferred in each case from a value assembly
25 line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated, which
30 computer program includes

- machine-readable instruction sequences of a first, higher-order hierarchy level, which prompt a computer to read in data from at least one data form and to combine these data using
35 specific combining rules, and to store the output data thus determined in a data form on a computer-readable storage medium and/or an output medium;

- 29 -

- 5 - mutually independent machine-readable instruction sequences of at least one further hierarchy level of lower order than the first hierarchy level, which instruction sequences prompt a computer to read out data at least from a number of data forms, to combine them with one another, and to store the results of combination in an output data form of this hierarchy level in each case; and
- 10 - machine-readable instruction sequences which prompt a computer to read data into input data forms from an input unit;
- 15 in which the data forms, which are being read on a specific hierarchy level during the program run, are output data forms of a hierarchy level of lower order than this hierarchy level, or input data forms:
- 20 in which all data forms have a standardized data format in such a way that all output data forms which are generated during the run of instruction sequences on an arbitrary hierarchy level, and all input data forms have a common data structure, that is to say data which are of one information type are always stored in the same position in a form;
- 25 and in which all different machine-readable instruction sequences of all different hierarchy levels are generated from identical source codes from which instructions for reading in the specific data forms and relating to the specific combinations are created by a specific parameter file either during generation of the sequences which can be executed, or during the running time of the computer program.

35

4. The computer program as claimed in claim 3, in which the data contained in the forms constitute input value packages (IVP) and value added packages (VAP) of the value assembly process.

5. The computer program as claimed in one of claims 3 or 4, in which the data generated by an instruction sequence are classified according to their qualitative information content and stored in different classes of standardized data forms.
6. The computer program as claimed in one of claims 3 to 5, in which during execution of the computer program or individual sequences of the computer program the generated value added packages (VAP) are compared with a reference value added package (RVAP) in each case, which reference value added package contains specification data, and in which a report is made via a warning function (EWS) in the event of impermissible specification deviations (NC).
7. The computer program as claimed in one of claims 3 to 6, in which machine-readable sequences of different value assembly subprocesses run on different computers, and data forms are transferred via long-distance data lines.
8. A method for visualizing a value assembly process on an output unit of a computer system, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which value assembly process value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated,

3906087-161903

5

10

15

20

35

with the aid of different colors and/or line thicknesses.

11. The method as claimed in one of claims 8 to 10, in
5 which a computer program in accordance with claim 6 runs in the central processing unit of the computer system, and in which the warning function prompts the computer system to display on the output unit an impermissible value deviation
10 occurring on an arbitrary hierarchy level, and all arrows which visualize value assembly lines which are affected by the value deviation are visualized in a particularly emphatic type of visualization on the output unit in such a way that the
15 impermissible deviation can be traced back immediately to its origin from a higher hierarchy level.
12. A computer program which includes instruction
20 sequences which prompt a computer to execute the method steps according to one of claims 8 to 11.
13. A computer system for carrying out a program
25 according to one of claims 3 to 7.
14. A computer program product in which instruction
sequences of the program are stored on a computer-readable medium in accordance with one of claims 3
30 to 7.
15. A computer program product in which instruction
sequences of the program are stored on a computer-readable medium in accordance with claim 12.